Computing Behaviours

	"Confidence in understanding problems and dealing with complexity"	"Persistence in working with difficult problems"	"Tolerance for ambiguity"	"Deals with open ended problems"	"Adapts existing knowledge or solutions to solve new problems"	"Iteratively develops, tests, and debug solutions"	"Weighs-up outcomes carefully"	"Communicates and works with others to achieve a common goal or solution"
Developing	● Makes no effort to attempt to grasp any aspects of the problem. ● Avoids problems/challenges that have more than one step or part to solving them. ● May unintentionally overly complicate problems	Doesn't engage with and avoids problems that are difficult to deal with and hard to solve.	● Refuses to acknowledge or struggles to accept ambiguity in both problems/solutions. ● Struggles to get started without a precise plan.	● Struggles to get started without a plan and clear expectations/deliverables. ● Follows instructions and only does what they are told.	Shows reluctance to or actively avoids learning from previous solutions.	● Struggles to express ideas as a solution. ● Looks to submit the first working solution as the finished product. ● Lacks an awareness of the need to test, debug and refine solutions iteratively. ● Has a preferred way of representing solutions and often chooses this method regardless of the task.	● Doesn't consider evaluating the efficacy of a solution.	● Has a negative impact on others. ● Communicates ineffectively ● Makes inappropriate contributions to the group. ● Uses incorrect subject vocabulary. Has no sense of own strengths and weaknesses or those of other.

Computing Behaviours

Emerging								
Emerging	• Gathers some of the necessary information to be able to understand the problem. • Grasps some but not all aspects of the problem often making educated guesses. • Consistently seeks advice and reassurance. • Requires support on how to approach each part of the problem/challenge.	Reluctantly engages with difficult problems but doesn't persevere for long.	Shows a willingness to acknowledge and accept some ambiguity exists in both problems/solutions.	• Does minimum expected but no more. • Investigates a limited number of problems /solutions.	• Struggles to identify patterns that match a problem to a previously learned solution. • When directed and with reassurance, will consider adapting pre-existing solutions to solve the current problem.	• Implements a solution using a given (completed) design. • Prototypes solutions quickly and submits more than one iteration of a solution for feedback. • Begins to use logical reasoning to predict outcomes. • Shows an awareness of the need to debug solutions but requires constant support and advice during this process. • When directed will consider different ways to represent solutions.	• Shows an awareness of the need to evaluate solutions against criteria but requires support and advice during this process.	• Does not have a negative impact on others. • Passively participates in the group, making no significant contribution. • Uses the correct subject vocabulary. • With prompts can explain how a solution works to others. • Has a sense of own strengths and weaknesses but evidences little strategy to deal with them.

Computing Behaviours

Secure								
	● Gathers all necessary information to understand the problem. ● Understands all aspects of a problem. ● Displays independence in breaking down problems and filtering out unnecessary information. ● With support, builds solutions in parts (sub-solutions) to recompose for a final solution.	● Responds positively to difficult problems and validates outcomes. ● Displays persistence at times of difficulty.	Shows an ability to tolerate ambiguity in both problems/solutions.	● Applies effort to independently explore an appropriate range of problems/solutions.	● Tinkers with solutions to find new uses. ● With some support identifies patterns and trends in problems and solutions. ● Chooses pre-existing solutions they are already aware of to adapt and solve the current problem.	● With support designs and models solutions. ● Requires occasional support when testing and debugging solutions. ● Uses logical reasoning to predict outcomes showing an awareness of inputs. ● With support chooses an appropriate way to represent solutions. ● Carefully records the iterative process and makes appropriate refinements to the solution.	● Independently evaluates the quality of solutions against given criteria. ● Considers if a solution is 'fit for purpose'. ● Views the work of others and identifies transferable efficiencies.	● Has a positive impact on the group. ● Communicates effectively with others. ● Groups working (with support) in parallel on the same problem/solution. ● Makes positive contributions to and supports others. ● Explains how a solution works to others. ● Understands own strengths and weaknesses and solicits help from appropriate others.

Computing Behaviours

Mastered								
	• Confidently and systematically breaks down problems and filters out unnecessary information and is able to explain the processes involved. • Effectively organises all parts of a problem to develop coherent sub-solutions. • Sub-solutions are appropriately coupled (joined). • Reduces the complexity of the solution without affecting its behaviour.	• Relishes difficult problems and finds fun in being persistent. • Able to explain how they have overcome difficult problems. • Demonstrates resilience despite setbacks.	• Celebrates ambiguity and having different interpretations • Compares the performance of interpretations that do the same thing.	Consistently displays curiosity to exhaustively investigate and analyse a broad range of appropriate problems/solutions.	• Independently identifies and acts on patterns in problems /solutions. • Independently seeks out pre-existing solutions (not directly within their existing sphere of knowledge or understanding) transferring ideas and/or solutions from one problem context to another.	• Independently designs, models, tests, debugs and refines solutions (using a test plan and data where appropriate). • Independently chooses an appropriate way to represent solutions. • Tidies up their solution to increase the comprehension of how it works. • Embeds comment and explanation of how a solution works to improve understandability and maintainability.	• Considers when good is good enough and uses this to design a criteria to critically evaluate the quality of a solution. • Measures the efficiency of solutions to improve them. • Makes trade-offs between conflicting demands then makes refinements and future solutions based on prioritising.	• Makes consistently positive contributions to the group. • Groups working effectively in parallel on the same problem/solution. • Balances autonomy and collaboration. • Values others' learning and teamwork styles to encourage contributions from others. • Where appropriate, constructively leads others. • Explains how a solution works with clarity. Uses team to effectively compensate for own weaknesses Uses team to effectively compensate for own weaknesses and uses own strengths to support others.

This work is based on the work of Thomas Stephens & Mark Dorling that can be found at http://code-it.co.uk/attitudes/